Madeleine Bates
Bolt Beranek and Newman, Inc.

Robert Ingria
Department of Linguistics, MIT

1. INTRODUCTION

This paper describes a sentence generator that was built primarily to focus on syntactic form and syntactic relationships. Our main goal was to produce a tutorial system for the English language; the intended users of the system are people with language delaying handicaps such as deafness, and people learning English as a foreign language. For these populations, extensive exposure to standard English constructions (negatives, questions, relativization, etc.) and their interactions is necessary. The purpose of the generator was to serve as a powerful resource for tutorial programs that need examples of particular constructions and/or related sentences to embed in exercises or examples for the student. The focus of the generator is thus not so much on what to express as on how to express it in acceptable English. This is quite different from the focus of most other language generation systems. Nonetheless, our system could be interfaced to a more goal-directed semantic component.

The mechanism of transformational grammar was chosen because it offered both a way to exercise tight control over the surface syntactic form of a sentence and a good model for the production of groups of sentences that are syntactically related (e.g. the active and passive forms of a transitive sentence). By controlling (at a very high level) the rules that are applied and by examining the detailed syntactic relationships in the tree structures at each end of the derivation, the tutorial part of the system accesses a great deal of information about the syntax of the sentences that are produced by the generator; this knowledge is used to give explanations and hints to the user in the context of the particular exercise that the student is attempting.

The transformational generator is composed of three major parts: a base component that produces base trees, a transformer that applies transformational rules to the trees to derive a surface tree, and a set of mechanisms to control the operation of the first two components. We will discuss each of the components of this system separately.

2. THE BASE COMPONENT

The base component is a set of functions that implicitly embody context free rules for creating a tree structure (phrase marker) in the X-bar framework (as discussed by Chomsky (1970), Jackendoff (1974), Bresnan (1975) and others.) In this system, the major syntactic categories (N(oun), V(erb), A(djective) and P(reposition)) are treated as complex symbols which are decomposable into the features [±N] and [±V]. This yields the following cross-classification of these categories:

		V: +	V: −
N	+	A	N
	−	V	P

Figure 1. Features in the X-bar System

The feature "N" marks a given category as "nounlike" (and thus corresponds to the traditional grammatical notion of "substantive") while "V" marks a category as "verblike." Nouns and Adjectives are [+N] because they share certain properties (e.g. Adjectives can be used in nominal contexts; in highly inflected languages, Adjectives and Nouns typically share the same inflectional paradigms, etc.) Adjectives and Verbs are [+V] because they share (among other things) various morphological traits (e.g. certain verbal forms, such as participles, have adjectival properties). Verbs and Prepositions are [-N] because they display common complement selection attributes (e.g. they both regularly take Nominal complements that bear Accusative Case.) (For further discussion of the issue of feature decomposition, and for some alternative proposals, see Jackendoff (1978) and George (1980a, Section 2; 1980b, Section 2).)

In addition, each syntactic category contains a specification of its rank (given in terms of number of bars, hence the term "X-bar" system). For instance, a Noun (N) is of rank 0 and is marked with no bars whereas the Noun Phrase which it heads is of the same category but different (higher) rank. Intermediate structures are also permitted; for instance, V′ (read "V bar") is that portion of the Verb Phrase which consists of a Verb and its complements (e.g. direct and indirect objects, clausal complements, prepositional phrases, etc.) while V′′ (read "V double bar") includes V′ as well as Auxiliary elements. For our purposes, we have adopted a uniform two-level structure across categories; that is, each category X is taken to have X′′ as its highest rank, so that Noun Phrase (NP) in our system is N′′, Verb Phrase is V′′, etc. Minor categories (such as DET(erminer), AUX(iliary), NEG(ative), etc.) stand outside this system, as do S(entence) and S′ (a sort of super sentence, which contains S and clause introducing elements (or "subordinating conjunctions") such as that). These categories are not decomposable into the features [±N] and [±V], and, except for S and S′, they do not have different ranks. (It should be noted that the adoption of a uniform two-level hypothesis and the placing of S and S′ outside of the normal X-bar system are not uncontroversial--see e.g. Jackendoff (1978) and George (1980a, Section 2; 1980b, Section 2). However, these assumptions are found in many variants of the X-bar framework and are adequate for our purposes.)

This work was sponsored by BEH grant #G007904514.

An example of the internal structure of the P'' corresponding to the phrase "to the sad boys" is given below:

```
P''  [ -V -N ]
  P'  [ -V -N ]
    P  [ -V -N ]
      to
    N''  [ +N -V PER.3 +DEF NU.PL
           +HUMAN GENDER.MALE ]
      DET  [ +DEF ]
        the
      A''  [ +N +V ]
        A'  [ +N +V ]
          A  [ +N +V ]
            sad
      N'  [ +N -V PER.3 +DEF NU.PL
            +HUMAN GENDER.MALE ]
        N  [ +N -V PER.3 +DEF NU.PL
             +HUMAN GENDER.MALE ]
          boy
```

Figure 2. Part of A Sample Base Structure

This system of cross-classification by features and by rank permits the creation of transformations which can refer to a specific rank or feature without referring to a specific major category. (See Bresnan (1975) for further discussion of this point.) For example, the transformation which fronts WH-words to form WH-Questions treats any X'' category as its target and, hence, can be used to question any of the major categories (e.g. A''--"how big is it?"; N''--"what did they do?" "which men left?"; P''--"to whom did you give it?"). Similarly, the transformation which marks Accusative Case on pronouns applies only to those N''s which follow a [-N] category; i.e. only to those N''s which are the objects of Verbs or Prepositions. This allows us to create extremely versatile transformations which apply in a variety of contexts, and frees us from the necessity of creating several transformations, each of which essentially replicates the Structural Description and Structural Change of the others, differing only in the category of the affected term.

A set of constraints (discussed further below) is the input to the base component and determines the type of base structure which is produced. A base structure has both the usual features on the nodes (category features such as [+N] and [-V], and selectional features such as [+PROPER]) and some additional diacritic features (such as [-C], for case marking) which are used to govern the application of certain transformations.

Lexical insertion is an integral part of the construction of the tree by the base component. It is not essential that words be chosen for the sentence at this time, but it is convenient because additional features in the structure (such as [+HUMAN], [+MALE]) are needed to guide some transformations (for instance, the insertion of the correct form of pronouns.)

In our current system, the choice of words to be inserted in the base structure is controlled by a dictionary and a semantic network which embodies a limited number of semantic class relationships and case restrictions to prohibit the production of utterances like "The answer saw the angry cookie." The network nodes are chosen at random for each sentence that is generated, but a more powerful semantic component could be used to convey particular "messages," provided only that it could find lexical items to be inserted in the small number of positions required by the base constraints.

3. THE TRANSFORMATIONAL COMPONENT

Each transformational rule has a Structural Description, Structural Change, and (optional) Condition; however rules are not marked as optional or obligatory, as they were in traditional transformational theory (e.g. Chomsky (1955)). Obligatory transformations whose structural descriptions were met would apply necessarily; optional transformations would apply at random. Moreover, various versions of transformational grammar have employed transformations as "filters" on possible derivations. In older work (e.g. the so-called "Standard Theory" (ST) of Chomsky (1965)) derivations in which a transformation required in a given syntactic configuration failed to apply would block, causing the result to be ruled out as ungrammatical (op. cit., p. 138).

In more recent theories (e.g. the "Extended Standard Theory" (EST) of Chomsky (1977) and Chomsky and Lasnik (1977)) all transformations are optional, freely ordered and may apply at random. Those derivations in which a transformation misapplies are ruled out by independent conditions on the structures produced by the operation of the transformational component (Chomsky (1977, p. 76)). These frameworks adopt a "generate and test" approach, wherein the misapplication of transformations during the course of a derivation (e.g. the failure of a required transformation to apply (ST, EST) or the application of a transformation in a prohibited syntactic configuration (EST)) will result in a rejection of this possible derivation. The application of different optional transformations results in the production of a variety of surface forms.

There are two reasons why we do not use this generate and test approach. The first is that it is computationally inefficient to allow the transformations to apply at random and to check the result to make sure that it is grammatical. More importantly, we view the transformations as tools to be used by a process outside the sentence generator itself. That is, an external process determines what the surface syntactic form of a given base structure should be; the transformations are not independent entities which make this decision on their own. For example, a focus mechanism should be able to select or prohibit passive sentences, a dialogue mechanism should be able to cause agent-deletion, and so on. In our application, tutorial programs select the characteristics of the sentences to be produced on the basis of the syntactic rule or rules being exercised in the particular tutorial.

The Structural Change of each transformation consists of one or more functions, analogous to the transformational elementaries of traditional transformational theory (Chomsky (1955, pp. 402-407, Section 93.1)). We have

not adopted the restriction on the Structural Change of transformations proposed by more recent work in generative grammar (e.g. Chomsky (1980, p. 4)) which prohibits "compounding of elementaries"; i.e. which limits the Structural Change of a transformation to a single operation. This would require breaking up many transformations into several transformations, each of which would have to apply in the derivation of a particular syntactic construction rather than having one transformation that performs the required operations. Inasmuch as we are interested in utilizing the generative capacity of transformational grammar to produce specific constructions, this break up of more general, overarching transformations into smaller, more specific operations is undesirable.

The operations that are performed by the rules are a combination of classic transformational operations (substitution, adjunction, deletion, insertion of non-lexical elements such as "there" and "do") and operations that linguists sometimes relegate to the base or post-transformational processes (insertion of pronouns, morphing of inflected forms). By making these operations rule-specific, many related forms can be produced from the same base tree and the control mechanisms outside the generator itself can specify which forms are to be produced. (Figure 3 shows some of the transformations currently in the system.)

```
SUBJECT-AUX-INVERSION

SD: (S´ (FEATS (TRANS.1)) COMP (FEATS (WH.+))
     1                          2
         (S N´´ TNS (OPT NODE (FEATS (M.+)))))
          3 4   5         6

SC:  (DeleteNode 6)
     (DeleteNode 5)
     (LChomsky 2 6)
     (LChomsky 2 5)

Condition: [NOT (EQ (QUOTE +)
                    (FeatureValue (QUOTE WH)
                                  (RootFeats 4]


RELATIVE-PRONOUN-SPELL-OUT  [REPEATABLE]

SD: (S´ XX (N´´ N´´ (S´ (COMP X (N´´
     1      2   3    4   5       6
                (FEATS (WH . +)) WH)))))
                                  7

SC:  (DeleteSons 6)
     (LSon 6 (if (EQ ´+(GetFeat 6 ´ HUMAN))
                 then ´who
                    else ´which))
```

Figure 3. Sample Transformations

Those transformations which affect the syntactic form of sentences are applied cyclically (see (Chomsky (1965, p. 143) for more details). Thus transformations apply from the "bottom up" during the course of a derivation, applying first in the most embedded clause and then working upwards until the matrix clause is reached. Within each cycle the transformations are strictly (and extrinsically) ordered. In addition to the cyclic syntactic transformations there exists a set of post-cyclic transformations, which apply after all the cyclic syntactic transformations have applied. These post-cyclic transformations, whose domain of operation ranges over the entire syntactic tree, supply the correct morphological forms of all lexical and grammatical items. This includes giving the correct plural forms of nouns, the inflected forms of verbs, the proper forms of pronouns (e.g. "he," "she" and "they" in subject position and "him," "her," and "them" in object position), etc. While it has been relatively rare in recent transformational analyses to utilize transformations to effect this type of morphological "spell-out," this mechanism was first proposed in the earliest work in generative grammar (Chomsky (1955)). Moreover, recent work by George (1980a; 1980b) and Ingria (in preparation) suggests that this is indeed the correct way of handling such morphological processes.

The transformations as a whole are divided up into "families" of related transformations. For example, there is a family of transformations which apply in the derivation of questions (all beginning with the prefix WH-); there is a family of morphing transformations (similarly beginning with the flagged mnemonic prefix MORPH-). These "families" of transformations provide detailed control over the generation process. For example, all transformations of the WH- family will apply to a single syntactic position that may be questioned (e.g. subject, direct object, object of preposition, etc.), resulting in questions of the form "Who died" and "To whom did she speak." This familial characterization of transformations is similar to the classical transformational approach (Chomsky (1955, p. 381, Section 90.1)) wherein families of transformations were first postulated, because of the condition imposed within that framework that each transformation must be a single-valued mapping.

Our current sentence generator produces declarative sentences, passive sentences (with and without agent deletion), dative movement sentences, yes-no questions and wh-questions (including multiple-wh questions such as "Who gave what to whom?"), there-insertion sentences, negated sentences (including both contracted and emphatic forms), relative clauses, finite and infinitival complements (e.g., "The teacher wanted Kathy to hurry."), imperative sentences, various complex auxiliaries (progressive, perfective, and modals), predicate adjectives, and predicate nominals. Although not all of these constructions are handled in complete generality, the generator produces a very large and natural subset of English. It is important to note that the interactions among all these transformations have been taken into account, so that any meaningful combination of them will produce a meaningful, grammatical sentence. (Appendix A lists some of the sentences which have been produced by the interaction of various transformations.)

In our application, there is a need to generate ungrammatical utterances occasionally (for example, in a tutorial exercising the student´s

ability to judge the grammaticality of utterances). To this end, we have developed an additional set of transformations that can be used to generate utterances which mimic the ungrammatical forms found in the writing of the language delayed populations for which this system is intended. For example, deaf and hearing-impaired children often have difficulty with negative sentences, and replace the not of Standard English negation with no and/or place the negative element in positions in which it does not occur in Standard English (e.g. "The mouse is no a big animal," "The girl no has gone," "Dogs not can build trees"). The fact that these ungrammatical forms may be modelled with transformations is highly significant, and lends support to the claim (Chapman (1974), Fromkin (1973)) that ungrammatical utterances are rule-driven.

4. HIGHER LEVELS OF CONTROL

In order to manage the creation of the base trees and the application of the transformational rules, we have developed several layers of control mechanisms. The first of these is a set of constraints that directs the operation of the base component and indicates which transformations to try. A transformational constraint merely turns a particular transformation on or off. The fact that a transformation is turned on does not guarantee that it will apply; it merely indicates that the Structural Description and Condition of that transformation are to be tried. Base constraints can have either atomic indicators or a list of constraints as their values. For example, the direct object constraint (DIROBJ (PER 3)(NU PL) ...) specifies all the base constraints necessary to produce the N'' subtree for the direct object position in the base structure.

There are a number of dependencies which exist among constraints. For example, if the transformational constraint for the passive transformation is turned on, then the base component must be instructed to produce a direct object and to choose a main verb that may be passivized; if the base constraint for a direct object is turned off, then the base constraint for an indirect object must be turned off as well. A data base of implications controls the application of constraints so that whenever a constraint is set (or turned off), the base and/or transformational constraints that its value implies are also set.

The notion of a particular syntactic construction transcends the distinction between base and transformational constraints. The "natural" specification of a syntactic construction such as passive or relative clause should be made without requiring detailed knowledge of the constraints or their implications. In addition, one might want to request, say, a relative clause on the subject, without specifying whether the target of relativization is to be the subject or object of the embedded clause.

We have developed a data base of structures called synspecs (for "syntactic specifications") which embody, at a very high level, the notion of a syntactic construction. These constructions cannot be identified with a single constraint or its implied constraints. (Implications specify necessary dependencies; synspecs specify possible but not necessary choices on the part of the system designers about what combinations of constraints should be invoked under a general name.) A synspec can contain an element of choice. The choice can be made by any user-defined function, though in our practice most of the choices are made at random. One example of this is a synspec called wh-question which decides which of the synspecs that actually set up the constraints for a wh-question (question-on-subject, question-on-object, question-on-dative, etc.) should be used. The synspecs also provide convenient hooks on which to hang other information associated with a syntactic construction: sentences exemplifying the construction, a description of the construction for purposes of documentation, etc. Figure 4 shows how several of the synspecs look when printed for the user.

```
wh-question

  Compute : (PickOne '(question-on-subject
                       question-on-object
                       question-on-dative))

  Description : (This SynSpec will create any
                 one of the questions with
                 WH-words.)


second-person-imperative

 BaseConstraints : ((IMPERATIVE . 2)
                    (TNS))

 TransConstraints :
           ((REQUEST-VOCATIVE-DELETION . +)
            (REQUEST-EXCLAMATION-INSERTION . +)
            (REQUEST-YOU-DELETION . +))

 Examples : ("Open the door!")
```

Figure 4. Sample SynSpecs

Synspecs are invoked through a simple mechanism that is available to the tutorial component of the system. Each tutorial specifies the range of constructions relevant to its topic and chooses among them for each sentence that is to be generated. To produce related sentences, the generator is restarted at the transformational component (using the previous base tree) after the synspecs specifying the relationship have been processed.)

Just as constraints have implications, so do synspecs. The relationships that hold among synspecs include exclusion (e.g. transitive-sentence excludes predicate-nominal-sentence), requirement (e.g. extraposed-relative requires relative-clause-on-subject or relative-clause-on-object), and permission (e.g. predicate-adverb-sentence allows there-insertion). A mechanism similar to the implications for constraints refines a set of candidate synspecs so that the user (or the tutorials) can make choices which are consistent. Thus the user does not have to know, understand, or remember which combinations of choices are allowed.

Once some constraints have been set (either directly or through synspecs), a command can be given to generate a sentence. The generator first assigns values to the constraints that the user did not specify; the values chosen are guaranteed to be compatible with the previous choices, and the implications of these choices ensure that contradictory specifications cannot be made. Once all constraints have been set, a base tree is generated and saved before the transformations are applied. Because the base structure has been saved, the transformational constraints can be reset and the generator called to start at the transformational component, producing a different surface sentence from the same base tree. As many sentences as are wanted can be produced in this way.

5. DEVELOPMENT TOOLS

As one side effect of the development of the generative system, we have built a debugging environment called the syntactic playground in which a user can develop and test various components of the generator. This environment has become more important than the tutorials in testing syntactic hypotheses and exploring the power of the language generator. In it, dictionary entries, transformations, implications and synspecs can be created, edited, and saved using interactive routines that ensure the correct format of those data types. It is also possible here to give commands to activate synspecs; this operation uses exactly the same interface as programs (e.g. tutorials) that use the generator. Commands exist in the playground to set base constraints to specific values and to turn individual transformations on and off without activating the implications of those operations. This allows the system programmer or linguist to have complete control over all aspects of the generation process.

Because the full power of the Interlisp system is available to the playground user, the base tree can be edited directly, as can any version of the tree during the derivation process. Transformations can also be "broken" like functions, so that when a transformation is about to be tried the generator goes into a "break" and conducts an interactive dialogue with the user who can control the matching of the Structural Description, examine the result of the match, allow (or not) the application of the Structural Change, edit the transformation and try it again, and perform many of the operations that are available in the general playground. In addition to the transformational break package there is a trace option which, if used, prints the constraints selected by the system, the words, and the transformations that are tried as they apply or fail. The playground has proved to be a powerful tool for exploring the interaction of various rules and the efficacy of the whole generation package.

6. CONCLUSION

This is the most syntactically powerful generator that we know of. It produces sets of related sentences maintaining detailed knowledge of the choices that have been made and the structure(s) that have been produced. Because the notion of "syntactic construction" is embodied in an appropriately high level of syntactic specification, the generator can be externally controlled. It is fast, efficient, and very easy to modify and maintain; it has been implemented in both Interlisp on a DECSystem-20 and UCSD Pascal on the Cromemco and Apple computers. It forms the core of a set of tutorial programs for English now being used by deaf children in a classroom setting, and thus is one of the first applications of computational linguistics to be used in an actual educational environment.

References

Bresnan, Joan (1975) "Transformations and Categories in Syntax," in R. Butts and J. Hintikka, eds. Proceedings of the Fifth International Congress of Logic, Methodology and Philosophy of Science, University of Western Ontario, London, Ontario.

Chapman, Robin S. (1974) The Interpretation of Deviant Sentences in English: A Transformational Approach, Janua Linguarum, Series Minor, Volume 189, Mouton, The Hague.

Chomsky, Noam (1955) The Logical Structure of Linguistic Theory, unpublished manuscript, microfilmed, MIT Libraries, partially published by Plenum Press, New York, 1975.

Chomsky, Noam (1965) Aspects of the Theory of Syntax, MIT Press, Cambridge, Massachusetts.

Chomsky, Noam (1970) "Remarks on Nominalization", in R. A. Jacobs and P. S. Rosenbaum, eds., Readings in English Transformational Grammar, Ginn and Co., Waltham, Mass.

Chomsky, Noam (1973) "Conditions on Transformations", in S. A. Anderson and P. Kiparsky, eds., A Festschrift for Morris Halle, Holt, Rinehart and Winston, New York.

Chomsky, Noam (1977) "On WH-Movement", in P. Culicover, T. Wasow and A. Akmajian, eds. Formal Syntax, Academic Press, Inc., New York.

Chomsky, Noam (1980) "On Binding," Linguistic Inquiry 11.

Chomsky, Noam and Howard Lasnik (1977) "Filters and Control", Linguistic Inquiry 8.

Fromkin, Victoria A. (1973) Speech Errors as Linguistic Evidence, Janua Linuarum, Series maior, Volume 77, Mouton, The Hague.

George, Leland M. (1980a) Analogical Generalization in Natural Language Syntax, unpublished Doctoral Dissertation, MIT.

George, Leland M. (1980b) Analogical Generalizations of Natural Language Syntax, unpublished manuscript, MIT.

Ingria, Robert (in preparation) Sentential Complementation in Modern Greek, Doctoral Dissertation, MIT.

Jackendoff, Ray S. (1974) "Introduction to the X´ Convention", distributed by Indiana University Linguistics Club, Bloomington.

Jackendoff, Ray S. (1978) X´ Syntax: A Study of Phrase Structure, Linguistic Inquiry Monograph No. 2, MIT Press, Cambridge, Mass.

Appendix A: Sample Sentences

1. Transitive Sentences

    1. The bullies chased the girl.
    2. What did the bullies do to the girl?
    3. They chased her.
    4. Who chased the girl?
    5. The bullies chased her.
    6. Who did they chase?
    7. Whom did they chase?
    8. They chased the girl.
    9. How many bullies chased the girl?
    10. Eight bullies chased the girl.
    11. How many bullies chased her?
    12. Eight bullies chased her.
    13. Who got chased?
    14. The girl got chased.
    15. She was chased by the bullies.
    16. The girl was being chased by the bullies.

2. Intransitive Sentences

    1. What did the girl do?
    2. She cried.
    3. Who cried?
    4. The girl cried.

3. Indirect Discourse

    1. Dan said that the girl is sad.
    2. Dan said that she is sad.
    3. Who said that the girl is sad?

4. Transitive Sentence with Indirect Object

    1. The generous boy gave a doll to the girl.
    2. The generous boy gave the girl a doll.
    3. The girl was given a doll.
    4. A doll was given to the girl.
    5. Who gave the girl a doll?
    6. Who gave what to whom?
    7. What did the generous boy give the girl?
    8. He gave her a doll.
    9. What did the generous boy give to the girl?
    10. He gave a doll to her.
    11. Who gave a doll to the girl?
    12. Who gave the girl a doll?
    13. Which boy gave the girl a doll?
    14. The generous boy gave her a doll.
    15. Which boy gave a doll to the girl?
    16. The generous boy gave it to her.
    17. How many dolls did the generous boy give the girl?
    18. He gave her one doll.

5. Comparative Sentences

    1. The soldier was better.
    2. The gentleman will be more unhappy.
    3. Alicia is hungrier than Jake.
    4. The children were angrier than Andy.

6. Superlative Sentences

    1. A policeman caught the nicest butterflies.
    2. A sheepdog was the sickest pet.
    3. The fire chief looks most generous.
    4. The smartest man swore.
    5. The oldest bulldog broke the dolls.

7. Sentences with Infinitives

    1. The teacher wanted Kathy to hurry.
    2. The gentleman promised the lady to close the door.
    3. The girls were hard to ridicule.

8. Relative Clauses

    1. Whoever embraced the kids will embrace the ladies.
    2. The girl who was intelligent cheated the adults.
    3. The woman who greased the tricycle mumbled.
    4. The teacher who lost the bulldogs swears.

9. Negative Sentences

    1. Kim won't help.
    2. Claire didn't help.
    3. The children won't shout.
    4. Do not slap the poodles.
    5. Do not cry.

10. Varieties of Quantifiers

    1. No toy breaks.
    2. Some excited boys kissed the women.
    3. Some hungry people eat.
    4. Two men cried.
    5. Every new toy broke.
    6. Not every man slips.
    7. The boy won't give the dogs any oranges.
    8. The girl doesn't see any cats.
    9. The old men didn't tell the boys any thing.
    10. The girl didn't love any body.

11. Varieties of Pronouns

    1. Bette is the sad one.
    2. Gloria is the happy one.
    3. Kevin is the saddest.
    4. Kathy is the most cheerful.
    5. Varda liked the sweet apple.
    6. Varda liked the sweet one.

12. THERE Sentences

    1. There were some toys in the dirt.
    2. There were no toys in the dirt.
    3. There weren't any toys in the dirt.